http协议

服务器把网页的html代码发送给浏览器，浏览器显示出来，浏览器和服务器之间的传输协议是HTTP

HTML是一种用来定义网页的文本，会HTML，就可以编写网页；

HTTP是在网络上传输HTML的协议，用于浏览器和服务器的通信。

Network显示浏览器和服务器的通信

http协议的分析

Network中，找到类型为html的记录，点击后右侧显示Request Headers（请求头）和Response Headers（响应头）



GET表示一个读取请求，将从服务器获得网页数据

HTTP/1.1指示采用的HTTP协议版本是1.1，1.1版本允许多个HTTP请求复用一个TCP连接，以加快传输速度。

200表示一个成功的响应，后面的OK是说明，返回的不是200，那么往往有其他的功能

http请求

1. 浏览器请求

方法：GET还是POST，GET仅请求资源，POST会附带用户数据；

路径：/full/url/path；

域名：由Host头指定：Host: www.baidu.com

以及其他相关的Header；

如果是POST，那么请求还包括一个Body，包含用户数据



如果一台服务器有多个网站，服务器就需要通过Host（请求网址）来区分浏览器请求的是哪个网站。

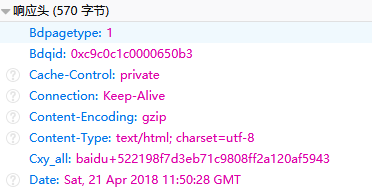
1. 服务器响应

响应代码：200表示成功，3xx表示重定向，4xx表示客户端发送的请求有错误，5xx表示服务器端处理时发生了错误；

响应类型：由Content-Type指定；

以及其他相关的Header；

通常服务器的HTTP响应会携带内容，也就是有一个Body，包含响应的内容，网页的HTML源码就在Body中。



HTTP响应分为Header和Body两部分（Body是可选项）

Content-Type指示响应的内容，这里是text/html表示HTML网页，浏览器依靠Content-Type来判断响应的内容是网页还是图片，是视频还是音乐，Body的数据类型由Content-Type头来确定，如果是网页，Body就是文本，如果是图片，Body就是图片的二进制数据。

浏览器并不靠URL来判断响应的内容，所以，即使URL是http://www.baidu.com/meimei.jpg，它也不一定就是图片。

HTTP响应的Body就是HTML源码

1. 如果浏览器还需要继续向服务器请求其他资源，比如图片，就再次发出HTTP请求，重复步骤1、2。

浏览器解析过程

当浏览器读取到首页的HTML源码后，它会解析HTML，显示页面，然后，根据HTML里面的各种链接，再发送HTTP请求给服务器，拿到相应的图片、视频、Flash、JavaScript脚本、CSS等各种资源，最终显示出一个完整的页面。所以我们在Network下面能看到很多额外的HTTP请求。

HTTP协议同时具备极强的扩展性，虽然浏览器请求的是http://www.baidu.com的首页，但是百度在HTML中可以链入其他服务器的资源，从而将请求压力分散到各个服务器上，并且，一个站点可以链接到其他站点，无数个站点互相链接起来，就形成了World Wide Web，简称WWW。

http格式

每个HTTP请求和响应都遵循相同的格式，一个HTTP包含Header和Body两部分，其中Body是可选的。

1. HTTP GET请求的格式，每个Header一行一个，换行符是\r\n。

GET /path HTTP/1.1

Header1: Value1

Header2: Value2

Header3: Value3

1. HTTP POST请求的格式：

POST /path HTTP/1.1

Header1: Value1

Header2: Value2

Header3: Value3

body data goes here...

当遇到连续两个\r\n时，Header部分结束，后面的数据全部是Body。

1. HTTP响应的格式：

200 OK

Header1: Value1

Header2: Value2

Header3: Value3

body data goes here...

HTTP响应如果包含body，也是通过\r\n\r\n来分隔的。

当存在Content-Encoding时，Body数据是被压缩的，最常见的压缩方式是gzip，所以，看到Content-Encoding: gzip时，需要将Body数据先解压缩，才能得到真正的数据。压缩的目的在于减少Body的大小，加快网络传输。

WSGI接口

底层代码由专门的服务器软件实现， Python专注于生成HTML文档，需要一个统一的接口，即WSGI：Web Server Gateway Interface。

WSGI要求实现一个函数，就可以响应HTTP请求

from wsgiref.simple\_server import make\_server #内置一个WSGI服务器  
#http处理函数  
def applicat(environ,response):  
 #environ为包含所有http请求信息的dict对象，response为一个发送http响应的函数  
 #调用response，发送http响应的header，header只能发送一次  
 #response函数接收两个参数，一个是http响应码，一个是一组list表示的HTTP header，每个header用一个包含两个str的tuple表示  
 response('200 OK',[('Content-Type','text/html')])  
 #body值返回给浏览器  
 body= '<h1>Hello, %s!</h1>' % (environ['PATH\_INFO'][1:] or 'web')  
 return [body.encode('utf-8')]  
#创建一个服务器，IP，端口和处理函数  
httpd = make\_server('', 8001, applicat)  
print('开始监听8001：')  
#开始监听HTTP请求  
httpd.serve\_forever()

web框架

用一个函数处理一个URL，Web框架来映射URL到函数

常见的Python Web框架有：

Django：全能型Web框架；

web.py：一个小巧的Web框架；

Bottle：和Flask类似的Web框架；

Tornado：Facebook的开源异步Web框架。

flask框架

Flask通过Python的装饰器在内部自动地把URL和函数给关联起来。

request.form['name']，来获取表单的内容。